

TypeScript 初体验 - 环境搭建与编译执行

TypeScript 初体验 - 环境搭建与编译执行

学习目标

环境搭建

安装 Node.js

安装 TypeScript 编译器

编写代码

编译执行

一些有用的编译选项

--outDir

--target

--watch

编译配置文件

指定加载的配置文件

学习目标

- 学会搭建 TypeScript 环境
- 掌握 TypeScript 代码的编译与运行

环境搭建

TypeScript 编写的程序并不能直接通过浏览器运行，我们需要先通过 TypeScript 编译器把 TypeScript 代码编译成 JavaScript 代码

TypeScript 的编译器是基于 Node.js 的，所以我们需要先安装 Node.js

安装 Node.js

<https://nodejs.org>

安装完成以后，可以通过 终端 或者 cmd 等命令行工具来调用 node

```
# 查看当前 node 版本
node -v
```

安装 TypeScript 编译器

通过 NPM 包管理工具安装 TypeScript 编译器

```
npm i -g typescript
```

安装完成以后，我们可以通过命令 tsc 来调用编译器

```
# 查看当前 tsc 编译器版本
tsc -v
```

编写代码

代码编辑器 - vscode

vsCode 和 TypeScript 都是微软的产品，vsCode 本身就是基于 TypeScript 进行开发的，vsCode 对 TypeScript 有着天然友好的支持

<https://code.visualstudio.com/>

TypeScript 文件

默认情况下，TypeScript 的文件的后缀为 .ts

TypeScript 代码

```
// ./src/helloKaIKEBa.ts
let str: string = '开课吧';
```

编译执行

使用我们安装的 TypeScript 编译器 tsc 对 .ts 文件进行编译

```
tsc ./src/helloKaIKEBa.ts
```

默认情况下会在当前文件所在目录下生成同名的 js 文件

一些有用的编译选项

编译命令 tsc 还支持许多编译选项，这里我先来了解几个比较常用的

--outDir

指定编译文件输出目录

```
tsc --outDir ./dist ./src/helloKaIKEBa.ts
```

--target

指定编译的代码版本目标，默认为 ES3

```
tsc --outDir ./dist --target ES6 ./src/helloKaIKEBa.ts
```

--watch

在监听模式下运行，当文件发生改变的时候自动编译

```
tsc --outDir ./dist --target ES6 --watch ./src/helloKaIKEBa.ts
```

通过上面几个例子，我们基本可以了解 `tsc` 的使用了，但是大家应该也发现了，如果每次编译都输入这么一大堆的选项其实是很繁琐的，好在 `TypeScript` 编译为我们提供了一个更加强大且方便的方式，编译配置文件：`tsconfig.json`，我们可以把上面的编译选项保存到这个配置文件中

编译配置文件

我们可以把编译的一些选项保存在一个指定的 `json` 文件中，默认情况下 `tsc` 命令运行的时候会自动去加载运行命令所在的目录下的 `tsconfig.json` 文件，配置文件格式如下

```
{
  "compilerOptions": {
    "outDir": "./dist",
    "target": "ES2015",
    "watch": true,
  },
  // ** : 所有目录（包括子目录）
  // * : 所有文件，也可以指定类型 *.ts
  "include": ["./src/**/*"]
}
```

有了单独的配置文件，我们就可以直接运行

```
tsc
```

指定加载的配置文件

使用 `--project` 或 `-p` 指定配置文件目录，会默认加载该目录下的 `tsconfig.json` 文件

```
tsc -p ./configs
```

也可以指定某个具体的配置文件

```
tsc -p ./configs/ts.json
```